

---

# LABORATÓRIO 7

---

TIPO CARACTERE E BOOLEANO

## EXERCÍCIOS DE FIXAÇÃO

---

VOCÊ DEVE FAZER OS EXERCÍCIOS PARA FIXAR O CONTEÚDO

1. Escreva uma função que receba um código numérico, tal como 65, e exiba na tela o caractere correspondente da tabela ASCII. No programa principal use essa função várias vezes para exibir os caracteres correspondentes aos números 80, 114, 111, 103, 67, 111, 109, 112.
2. Escreva um programa que dê um sinal de alerta sonoro e peça uma senha numérica. O programa deve usar uma função para emitir o sinal de alerta e outra função para ler e retornar a senha. A função principal deve se encarregar das demais tarefas. Use o caractere sublinhado para indicar o local (e a quantidade de dígitos) da senha.

Iniciando com um som...  
Digite sua senha: \_\_\_\_\_  
Obrigado!

3. Escreva uma função chamada `charada()` que receba um caractere `ch` e um número `x`. A função deve retornar o caractere que está `x` posições à frente de `ch` na tabela ASCII. A função principal deve usar `charada()` várias vezes para exibir a palavra “Papel”. Isso deve ser feito com cinco chamadas à função, com cada uma das chamadas usando os respectivos valores para `x`: 20, 90, 72, 48, 107.

**Dica:** procure na tabela ASCII os caracteres apropriados para passar como argumento de forma a obter a palavra “Papel” como resultado.

4. Escreva um programa que use a instrução `cout` juntamente com o caractere de tabulação para produzir a seguinte saída:

```
Maça =      20 Unidades
Melancia =   50 Unidades
Ameixa =     30 Unidades
```

5. Escreva as funções `ligarBit`, `deligarBit` e `testarBit`. Elas devem receber um `unsigned char` e a posição do bit. A função `TestarBit` deve retornar um booleano indicando o estado do bit e as demais devem retornar o valor `unsigned char` modificado.

## EXERCÍCIOS DE APRENDIZAGEM

---

VOCÊ DEVE ESCREVER PROGRAMAS PARA REALMENTE APRENDER

1. Crie uma função chamada `codificar` que recebe um caractere, soma 3 a ele e retorna o novo caractere codificado. Crie outra função chamada `decodificar` que recebe um caractere e subtrai 3 do caractere recebido e retorna o novo caractere decodificado. Use as duas funções em um programa que receba um caractere do usuário, codifique o caractere, mostre o caractere ao usuário, decodifique-o e mostre-o novamente.
2. Usando as funções criadas na questão anterior, peça ao usuário para digitar uma letra, e posteriormente pergunte se ele quer decodificar ou codificar a letra recebida, pedindo para que ele entre com 1 se desejar codificar e com 0 se desejar decodificar. Utilize uma estrutura `if else` como exemplificado abaixo:

```
if (resposta == true)
    codificar(ch);
else
    decodificar(ch);
```

3. Construa uma função `bitsBaixos` que utiliza uma máscara de bits para retornar apenas os 16 bits de mais baixa ordem de um número tipo `int`. Teste a função passando um valor inteiro digitado pelo usuário e mostrando o número retornado.

```
Digite um valor inteiro: 350000
Os 16 bits mais baixos desse valor correspondem ao número 22320.
```

4. Construa uma função `bitsAltos` que utiliza uma máscara de bits para retornar apenas os 16 bits de ordem mais alta de um número tipo `int`. Teste a função passando um valor inteiro digitado pelo usuário e mostrando o número retornado.

```
Digite um valor inteiro: 350000
Os 16 bits mais altos desse valor correspondem ao número 5.
```

5. Construa uma função `exibirBits` que, dado um valor `unsigned char`, exibe na tela a representação binária desse número. A função `exibirBits` deve usar a função `testarBit` (do exercício de fixação 5) para descobrir quais bits do valor recebido são iguais 0 e quais são iguais a 1.

```
Digite um valor entre 0 e 255: 143
O número 143 em binário é 10001111.
```